Use of Ontology for Solving Interoperability Problems between Enterprises

Hui Liu[1,2], Anne-Françoise Cutting-Decelle[1,2], Jean-Pierre Bourey[1,2]

[1] Univ Lille Nord de France, F-59000 Lille, France
[2] LM2O, Ecole Centrale de Lille
Cité Scientifique – BP 48 – 59651 Villeneuve d'Ascq Cedex
{hui.liu, anne-francoise.cutting-decelle, jean-pierre.bourey}@ec-lille.fr

Abstract. When trying to solve interoperability problems between enterprises, the semantic issues are important. To date, they are more and more focused on ontology. This paper presents how to use ontology in the PBMEI method, aimed at solving enterprise interoperability problems in modelling environment. During the elaboration of PBMEI, the necessary ontology information is explicitly specified. Because of two different uses of ontologies, this paper proposes two variants of PBMEI. Finally, this paper concludes with the content of the ontologies required in PBMEI. The ontologies in the PBMEI method PBMEI for an application case are being studied.

Keywords: Ontology, collaborative process, interoperability process, transformation

1   Introduction

How to bring together the distributed and heterogeneous information systems of enterprises and make them collaborate with each other to achieve a certain business objective is known as the enterprise interoperability [1] problem. To solve this kind of problem, data heterogeneity must be considered from two aspects: structural/schematic heterogeneity and semantic heterogeneity. Structural heterogeneity may be caused by type conflicts, labeling conflicts, aggregation conflicts, and generalization conflicts occurring in different databases/information systems [3]. This kind of heterogeneity can be solved by various techniques, such as XML, SOAP, EAI, ESB/SOA among others; semantic heterogeneity may come from naming conflicts, scaling and unit conflicts and confounding conflicts in different systems [3]. To overcome the problem of semantic heterogeneity, ontologies and semantics-based technologies can play a key role [4, 5]. How to use ontology in integration systems to achieve semantic interoperability is studied in [5] from four main criteria: role/architecture of ontology, ontology representation [8, 9], use of mapping [10] and ontology engineering [11], and furthermore in [7], semantic interoperability is discussed in more detail from the point of view of ontology mapping in three aspects: mapping discovery, mapping representation and reasoning with mapping. However all the above researches deal with the way of using ontology during the integration process, but how to use ontology in the modelling phase of an

integration system, especially for enterprise interoperability remains a problem. This problem will be discussed here based on the method we proposed in [6].

The proposed method in [6] aims at solving the enterprise interoperability problem. It starts from modelling collaboration requirements between enterprises with collaborative process[1] and after several steps of transformation, it ends up with executable interoperability processes, such providing a method for the modelling environment. In [6], the use of ontology has not been discussed. This paper will discuss how to use ontology in our proposed method to solve enterprise interoperability problem.

So far the method proposed in [6] has been enhanced and it will be elaborated in Section 2. According to different uses of ontologies in our method, the proposed method has two variants, which are discussed in Section 3. Section 4 concludes with the content of ontologies.

2   Process-Based Method for Enterprise Interoperability

In order to solve enterprise interoperability problem, a "Process-Based Method for Enterprise Interoperability" (PBMEI) is proposed in the paper and illustrated in Fig. 1. In PBMEI, business requirements about enterprise interoperability are represented in collaborative processes among which the enterprises involved. Through the analysis of the development mode of information systems [15], the limitation of workflows [16] and the advantages of web services (WS) [17], our method focuses on service-related process specification languages to describe collaborative processes; for example BPMN [13]. The collaborative process will finally be achieved through interoperability processes which are described in executable process specification languages. Our method is process-based and it also makes relevant enterprises more responsive to changing collaboration requirements. PBMEI is described in detail in the following sections.
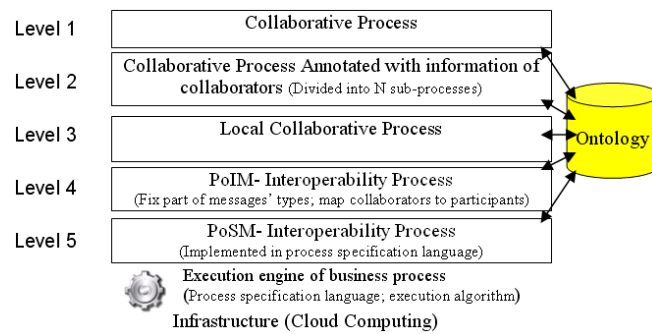

Fig. 1. Process-Based Method for Enterprise Interoperability

At the first level, collaborative process must be defined from two aspects: business flow and data/message flow, which is inspired by [2]. According to [2], US Army

---

[1] The definitions about collaborative process, interoperability process, collaborator, participant, cooperator, internal process, coordination process, cooperation process can be found in [6].

proposed an expansion of the system architecture into three further sub-architectures: software architecture, data architecture and network architecture. Software architecture defines the functionality of each module, data architecture is related to data definition, and network architecture is related to software deployment requirement. Furthermore, all business requirements must be mapped onto a certain system architecture to be implemented. The collaborative process is one kind of business requirements, so collaborative process must also be mapped onto the above three sub-architectures, that is to say a collaborative process must have some aspects that can be mapped onto the above three sub-architectures. However, the network sub-architecture is determined by concrete business requirements and related to the whole system, so this paper will consider this problem from the overall point of view, not in collaborative processes. Finally the collaborative process will be constructed from two aspects: functionality and data. The business flow describes the functionality of the collaborative process and the data flow describes the data exchanged in the process.

At the second level, the collaborative process will be annotated with collaborators' information, i.e., all the activities in the process must be charged by one collaborator. This task depends on an ontology. When searching the relevant collaborators for a certain activity, the ontology will be inspected to determine which collaborator can do such activity. If several candidates are selected, the target candidate will be selected according to the collaboration policy/requirements, or according to predefined conditions, such as QoS, trust rank/belief value etc. So the ontology must contain the necessary information about all the collaborators (for example, collaborator's name, historical information about service running, responsibility, etc). Once annotated with the information of collaborators, the collaborative process will be transformed into a set of collaborative sub-processes as long as it follows a given transformation method.

At the third level, the collaborative processes (including the generated collaborative sub-processes) will be transformed into local collaborative processes by each collaborator. During this transformation, the business terminologies will be transformed from global to local terminologies and the process specification language will also be transformed from global to local if necessary.

At the fourth level "PoIM" (**Pro**tocol **I**ndependent **M**odel), messages type in collaborative process must be determined according to messages context (messages sender and receiver, and relevant business context). Some messages type information may also be partially declared in the collaboration requirement. The above two cases of message type determination are ontology-based. This ontology must contain the definitions of business messages which have some context specifications.

After the determination of the type of messages, collaborators in a collaborative process will be mapped onto participants. The key of the mapping focuses on an activity functionality and context. After the mapping from collaborator to participant, the collaborator's information in the process must also be kept, since such information has a semantics that is not included in participants, for example, several roles of collaborators, whose semantics cannot be represented by the participant roles. The above tasks also rely on ontology. And because a participant is an element of the system architecture, the ontology must also contain the information about each collaborator's system architecture. At last, after messages' types are fixed and the

mapping from collaborators to participants is done, collaborative processes become interoperability processes.

At the fifth level "PoSM" (**Pro**tocol **S**pecific **M**odel), the interoperability process will be implemented in an executable process specification language, and all the message transport protocols will be explicitly specified.

According to this description, PBMEI closely depends on ontology and SOA and it also has one precondition: interoperability process totally depends on the original functions of each collaborator's information system. Of course PBMEI also relies on the process execution engine and a given infrastructure, such as cloud computing infrastructure [18, 19].

3   Two variants of PBMEI

In [6], collaborative business processes are divided into three types: internal process, coordination process, and cooperation process. As internal process and coordination process can be easy to implement with the help of WS-Business Process Execution Language or workflow model, this paper focuses on cooperation process in PBMEI.

When using PBMEI to solve interoperability problems, the first problem we meet is the following: who will create the cooperation process and in which style? In practice, if there is a core cooperator, the cooperation (collaborative) process is created by the core cooperator who will not negotiate with any other cooperator; if there is no core cooperator, the cooperation (collaborative) process is created through the negotiation of all the cooperators. When applying PBMEI to the above two cases, two variants of PBMEI are generated -- they are described below.

3.1   Process-Based Method for collaboration without core cooperator

If the collaborative process has no core cooperator, PBMEI becomes the following variant, see Fig. 2. The first and second levels and the transformation between them are global and depend on the global ontology; the third, fourth and fifth levels and the transformations related to them are done locally by each separate collaborator, depending on the local ontology. Once all the collaborators have generated their own interoperability process, they can execute it through an identical execution algorithm.

The global ontology is created, managed and accessed by all cooperators, and it includes the common sense necessary when the collaborators negotiate with each other to create a collaborative process. The global ontology will also provide the syntax and the semantics of the collaborative processes. The business expressions of all elements of a collaborative process must also respect the definitions in the global ontology. The global ontology must also contain the collaborators' information needed when the collaborative process is annotated.

Each cooperator creates and manages its own local ontology which can be accessed by the other cooperators under a certain condition. The local ontology contains all the information about the enterprise architecture for a corresponding collaborator. It includes three basic ontologies: business ontology, model ontology and network ontology. Business ontology contains all the terminology related to local business

requirements. Model ontology contains all the models (architecture models and data models) corresponding to the different software development phases. Network ontology contains the information about software deployment.

Of course, there must be mappings between the global ontology and the local ontologies for all collaborators and such mappings will be used by each collaborator to transform the collaborative process into their own collaborative process. The mapping between the global and local ontology will be stored and maintained in the corresponding local ontology for each cooperator.
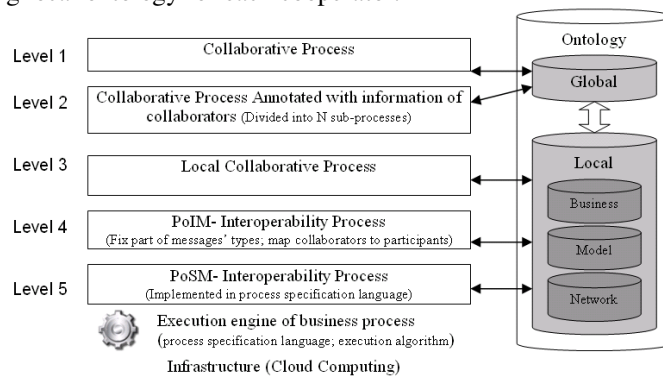

Fig. 2. Process-Based method for collaboration without core cooperator

3.2   Process-Based Method for collaboration with core cooperator

If a collaborative process in PBMEI has a core cooperator, once the core cooperator generates the process, it transforms the process into executable interoperability processes and then deliver them to its collaborators. The other collaborators will then transform the received processes into processes which are expressed in their own languages and based on their own local ontologies.
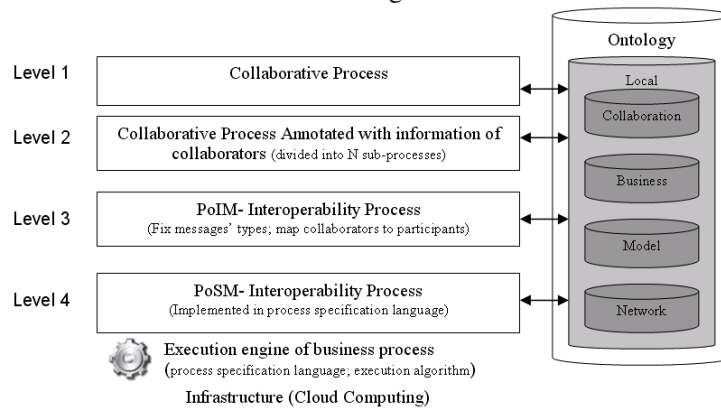

Fig. 3. Process-Based method for collaboration with core cooperator

Since the collaborative process is created by the core cooperator, the process is only based on the core collaborator's local ontology, and there is no need to transform the global collaborative process into local collaborative process: this is why the variant depicted in Fig. 3 does not have the level "Local collaborative process".

Note that besides the business ontology, model ontology and network ontology, the ontology of the core cooperator also contains the collaboration ontology which offers information about the other collaborators and their services.

4   Content of the ontologies in PBMEI

As Section 2 proposes suggestions about the content of the ontologies in PBMEI, and Section 3 provides the categories of ontologies in PBMEI, this section will present the content of each ontology: see Table 1. Fig. 4 shows an example of the global ontology in PBMEI, as created using Protégé v3.4.3[20].

Table 1.   Content and mapping of ontologies in PBMEI

Ontology		Content	Mapping
Local / Global	Global	• information about all the collaborators: name, business roles, and postal address, email address, network address, offered business services, published web services and related statistical information about their offered services (e.g., QoS, trust rank), etc • common business object model • specification language for collaborative processes and business policies • collaborative process, collaboration policies	
	Collaboration	• information about all the collaborators: name, business roles, and postal address, email address, network address, offered business services, published web services and related statistical information about their offered services (e.g., QoS, trust rank), etc • information about services published by other collaborators • collaborative process, collaboration policies	
	Business	• all business concepts in its own domain and relationships • local business policies • local collaborative process • organizational information	Mapping to/ from global ontology; Mapping to model ontology;
	Model	• formal descriptions of business concepts, their relationships • formal descriptions of business rules and technical rules • formal descriptions at different levels about the architecture of an enterprise information system • formal descriptions about all physical components of enterprise software systems • Interoperability processes at "PoIM" and "PoSM" levels	Mapping to business ontology; Mapping to network ontology;
	Network	• deployment information of all software components in an enterprise information system	Mapping to model ontology

Table 1 shows mappings between global/collaboration ontology and business ontology, between business and model ontology and between model and network ontology. These mappings must be maintained and managed and they will be used during the transformation of business processes. The above mappings between ontologies of a collaborator are vertical, but there are also the horizontal mappings between ontologies of different collaborators, and such mappings will be used in the second variant of PBMEI, which will be presented in another paper.
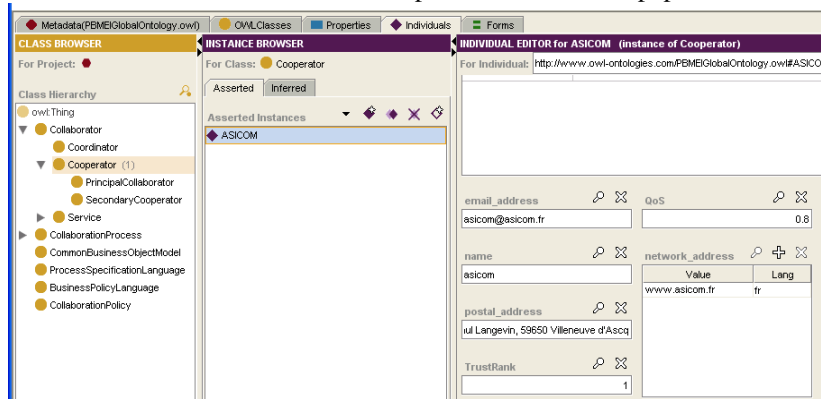

Fig. 4. Example of the Global Ontology in PBMEI

According to Table 1, the ontology contains information about collaboration, business, model and deployment. In fact, the data storage in the ontology can be real or virtual. This means that data can be directly stored in the ontology, or they can also be stored in remote professional servers : in that case the ontology only stores the ontology-based description of remote data, for example, business rules can be stored in Business Rule Management System (BRMS) [14]. Theontology only contains ontology-based description of business rules.

5   Conclusion

This paper presents some initial developments about the PBMEI method which uses ontology in modelling environment to solve enterprise interoperability problems. This method also makes collaborators in collaborative process easily adaptable to collaboration requirement changes. After analyzing the dependent information in PBMEI and its two variants, the global ontology, local ontology, business ontology, model ontology and network ontology are introduced and a first overview of their contents is proposed, thus building the foundation for further research on the way of using PBMEI in a concrete application case.

The proposed PBMEI method is ontology-based, process-based and model-driven [12] and it is also ontology-language-independent. However there are still many research points to be done in the future, for example, how to validate whether a collaborative process is based on a designated ontology or how to support process transformation according to horizontal ontology mappings between collaborators.

References

1. Chen, D., Doumeingts, G.: "European Initiatives to develop interoperability of enterprise applications - basic concepts, framework and roadmap". Journal of Annual reviews in Control, 27 (3), 151-160, Elsevier (2003)
2. Hamilton, J.A.: A practical application of enterprise architecture for interoperability. ISE '03, ISBN: 1-56555-270-9 (2003)
3. Goh, C.H.: Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources. Phd, MIT (1997)
4. Uschold, M., Grüninger, M.: Ontologies and semantics for seamless connectivity. SIGMOD Record, 33(3) (2004)
5. Wache, H.: Ontology-Based integration of information, a survey of existing approaches. workshop on ontologies and information sharing, IJCAI (2001)
6. Liu, H., Bourey, J.-P.: Transformation from a Collaborative Process to Multiple Interoperability Processes. I-ESA Conference, Coventry (2010)
7. Noy, N.: Semantic Integration: A Survey of Ontology-based Approaches. Sigmod Record, Special Issue on Semantic Integration (2004)
8. Corcho, O., Gómez-Pérez, A.: Evaluating knowledge representation and reasoning capabilities of ontology specification languages. In Proceedings of the ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin (2000)
9. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The Making of a Web Ontology Language. J. of Web Semantics, 1(1):7-26 (2003)
10. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. The Knowledge Engineering Review, 18(1):1–31 (2003)
11. Gómez-Pérez, A.; Fernández-López, M.; Corcho, O.: Ontological Engineering. Springer, Berlin (2004)
12. Bourey, J.-P., Grangel Seguer, R., Doumeingts, G., Berre A.-J., Deliverable DTG2.3 Report on Model-Driven Interoperability. (2007) http://interop-vlab.eu/ei_public_deliverables/interop-noe-deliverables/dap-domain-architecture-and-platforms/D91/
13. OMG:Business Process Model and Notation. http://www.omg.org/spec/BPMN/1.2/ (2009)
14. Graham, I.: Service Oriented Business Rules Management Systems. (2005) http://www.trireme.com/whitepapers/Business%20rules/Trireme_Report_Service_Oriented_Business_Rules_Management_Systems_Ver2b.pdf
15. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Process-Aware Information Systems: Bridging People and Software through Process Technology. Wiley& Sons, (2005)
16. van der Aalst, W M. P., Benatallah, B., Casati, F., Curbera, F., Verbeek, E.: "Business process management: Where business processes and web services meet". Data Knowl. Eng. 61(1): 1-5 (2007)
17. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services Concepts, Architectures and Applications. Springer-Verlag, Berlin (2004)
18. Chappell, D.: Introducing the Azure services platform. DavidChappell&Associates (2008). http://download.microsoft.com/download/e/4/3/e43bb484-3b52-4fa8-a9f9-ec60a32954bc/Azure_Services_Platform.docx
19. Rimal, B.P.; Eunmi Choi; Lumb, I.: "A Taxonomy and Survey of Cloud Computing Systems". INC, IMS and IDC, 2009, NCM '09, Fifth International Joint Conference on, vol., no., pp.44-51, 25-27 (2009)
20. Protégé, http://protege.stanford.edu/